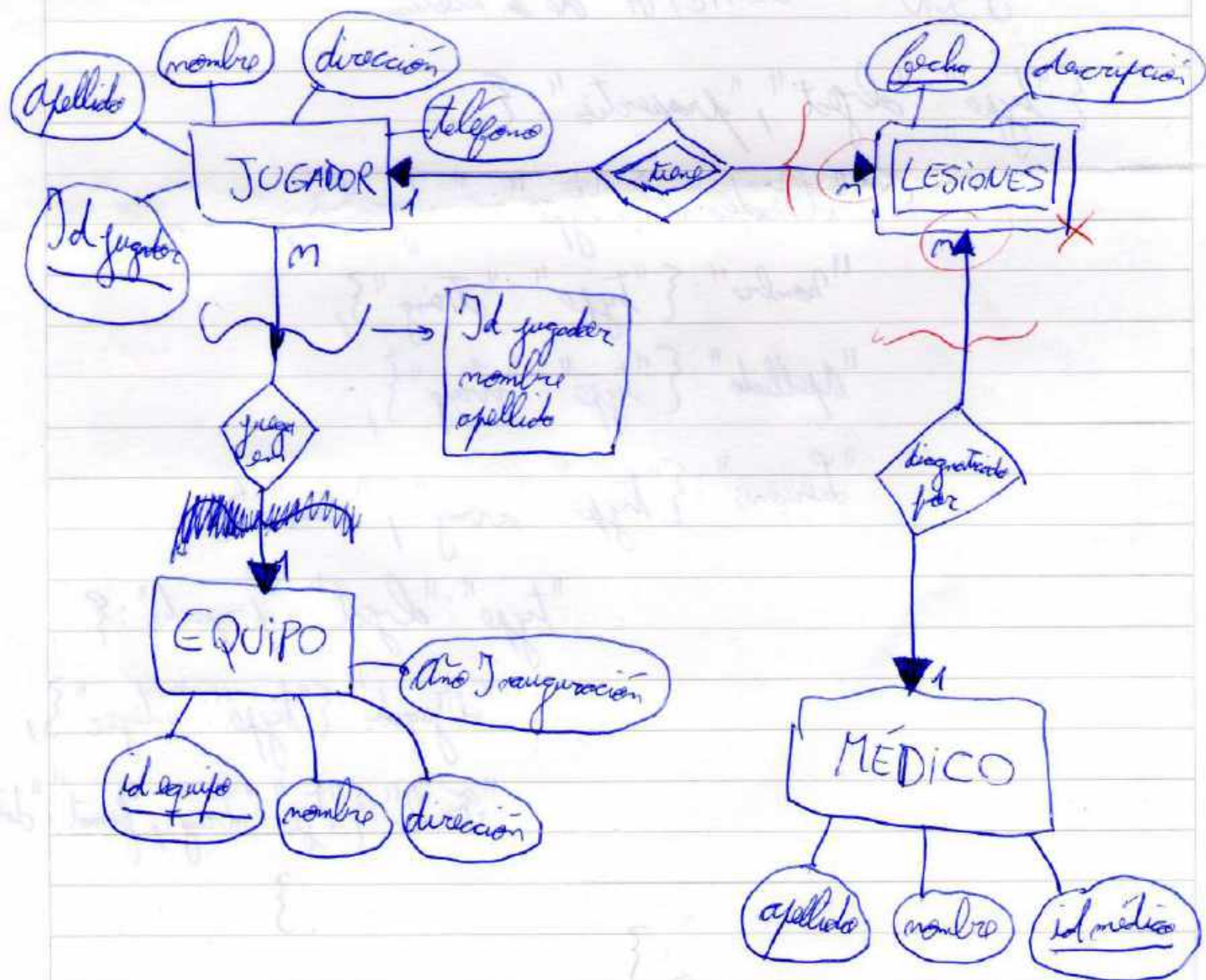


1) a) Para que un equipo pueda obtener rápido a un jugador, voy a realizar una incrustación de estos mismos en equipo. Pero no de sus atributos "id jugador", "nombre" y "apellido", que es lo que especifica el enunciado.

Como es tan importante obtener la lesión de un jugador o nombre de un médico, voy a utilizar referencias en el caso para que se pueda acceder a la información.

¡No! ES DÉBIL



- Cada jugador tiene una referencia al equipo en que juega
- Cada jugador tiene una referencia a su lesión (un array de lesiones)
- Cada lesión tiene una referencia al médico que la diagnostica
- Cada lesión tiene una referencia al jugador correspondiente (y esta es obligatoria ya que las lesiones son propiedad del jugador)
- Cada médico tiene un array de referencias a las lesiones que atendió, y esas referencias contienen el id jugador y la fecha correspondiente para que pueda ser identificado.
- Cada equipo tiene incrustado un arreglo de sus jugadores, nombre, apellido > con el cual identifica a sus jugadores.

### JSON SCHEMA de Médico

```

{ "type": "object", "properties": {
  "idMédico": { "type": "integer" },
  "nombre": { "type": "string" },
  "apellido": { "type": "string" },
  "Lesiones": { "type": "array", "items": {
    "type": "object", "properties": {
      "idJugador": { "type": "integer" },
      "Fecha": { "type": "string", "format": "date-time" }
    }
  }
}
}

```

b) i)

DATOS JUGADORES x EQUIPOS	
AÑO INAUGURACIÓN	K
ID EQUIPO	C↑
ID JUGADOR	C↑
NOMBRE	
APELLIDO	

MR 1

MR 3 (8)

MRS

MRS

8 a MR3 porque a una brigada por desigualdad (al estar buscando en un rango)

→ NO, PONÉS UNA CUBE DUMMY Y ORDENÁS POR AÑO

ii)

DATOS EQUIPOS	
DUMMY	K
NOMBRE	C↑
ID EQUIPO	C↑
DIRECCIÓN	
AÑO INAUGURACIÓN	

MR 1

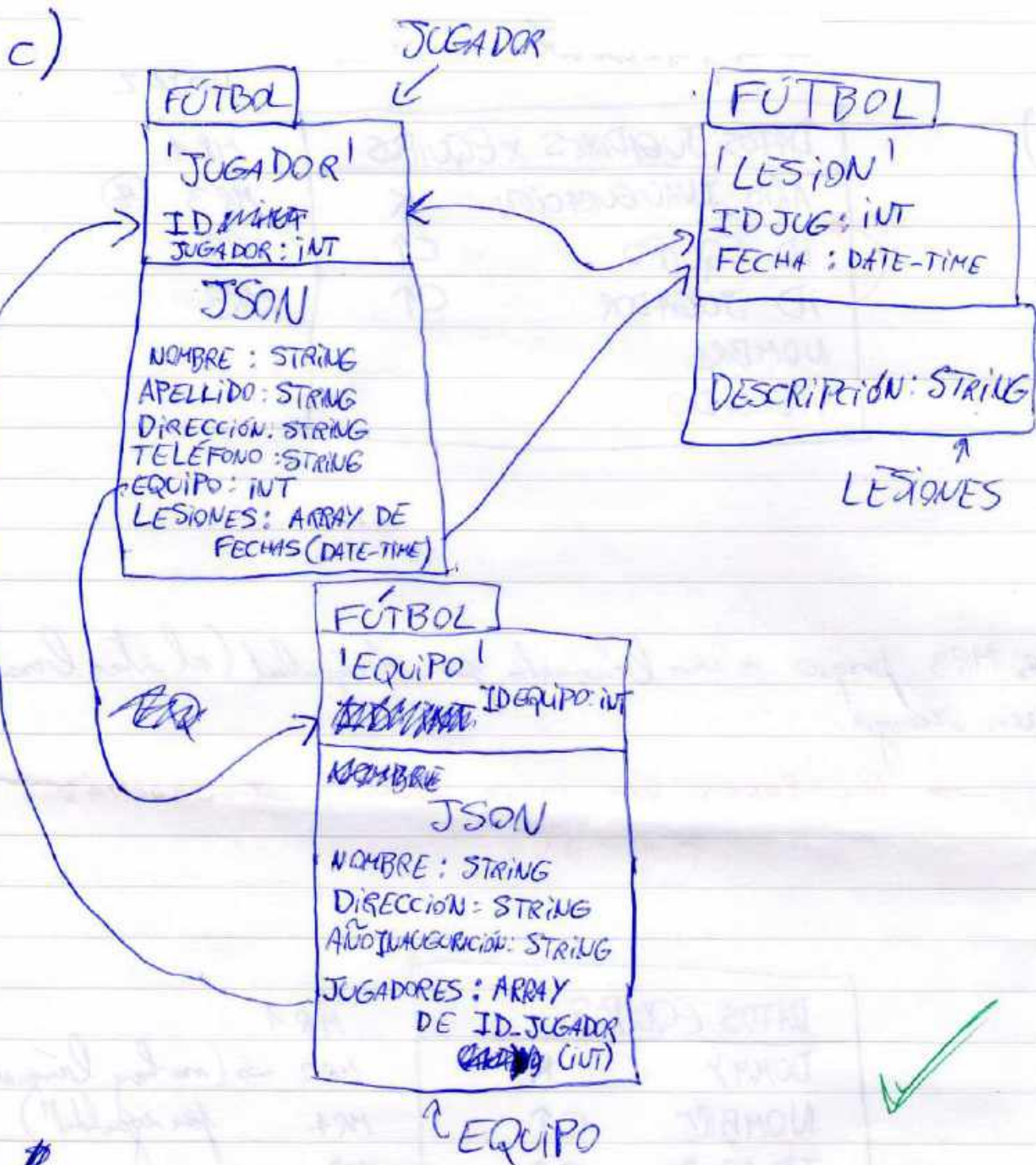
MR 2 → (no hay brigada por desigualdad)

MR 4

MRS







Desde jugador puede obtener la fecha en la que sufrió una lesión, y con ese más su ID puede acceder a cada lesión.  
 Basta de la clave de cada lesión es el ID del jugador, con lo que puede obtener al jugador correspondiente de la lesión.  
 Además cada jugador tiene un INT EQUIPO con

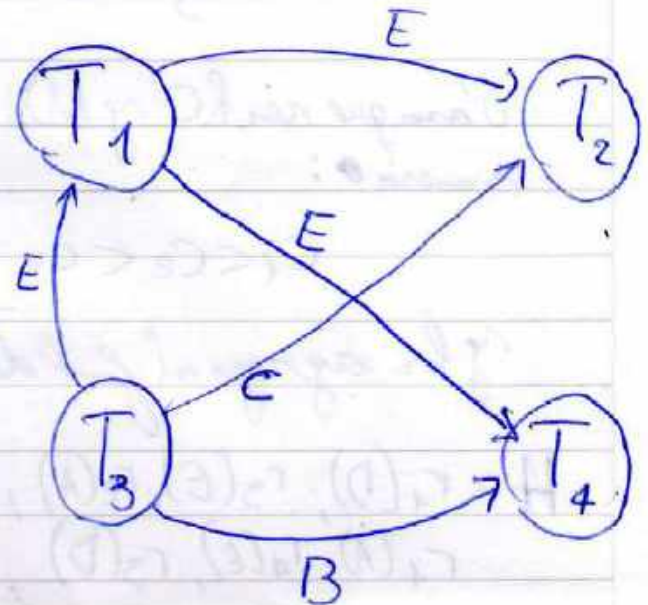
Además, desde cada jugador puede acceder a su equipo desde "EQUIPO" utilizando su clave.  
 Es un equipo puede acceder a sus jugadores gracias a "JUGADORES", un arreglo/lista de INTS que son los IDs de cada jugador.



2) a) Ponga la historia de forma tabular

$T_1$	$T_2$	$T_3$	$T_4$
$r(D)$			
		$r(E)$	
	$r(A)$		
$w(E)$		$w(B)$	
	$r(E)$		
	$w(C)$	$r(C)$	
			$r(B)$
$r(A)$			$r(E)$
		$r(D)$	

i)



Como el grafo es acíclico,  
es SR

$\left. \begin{matrix} T_3 T_1 T_2 T_4 \\ T_3 T_1 T_4 T_2 \end{matrix} \right\}$  son las historias  
seriales equivalentes

ii) Para que sea RC: si  $T_i$  lee de  $T_j$  (o sea  $T_j$  fue la última transacción en escribir  $X$  que no abortó, y luego  $T_i$  lee  $X$  (para algún  $X$ )) entonces  $c_j < c_i$

Para que sea ACA: si  $T_i$  lee de  $T_j$ , entonces  $T_j$  tuvo que haber commitado antes de que  $T_i$  leyera

Como donde una transacción lee a otra:  $T_4$  lee  $B$  de  $T_3$

-  $T_4$  lee  $B$  de  $T_3$

-  $T_2$  lee  $E$  de  $T_1$

-  $T_4$  lee  $E$  de  $T_1$



Para que sea RC entonces

$$C_3 < C_4 \quad C_1 < C_2 \quad C_1 < C_4$$

Para que sea ACA

$$C_3 < r_4(B) \quad C_1 < r_2(E) \quad C_1 < r_4(E)$$

Para que sea RC y NO ACA, ordena los commits de la siguiente manera:

$$C_1 < C_2 < C_3 < C_4$$

y los agregas al final de la historia

$$H = r_1(D), r_3(E), r_2(A), w_3(B), w_1(E), r_2(E), r_3(C), w_2(C), r_4(B), \\ r_1(A), r_4(E), r_3(D), C_1, C_2, C_3, C_4$$

Para que sea ACA los ordenas

$$C_3 < C_1 < C_2 < C_4 \text{ y los colocas de la siguiente manera}$$

$$H = r_1(D), r_3(E), r_2(A), w_3(B), C_3, w_1(E), C_1, r_2(E), r_3(C), \\ w_2(C), C_2, r_4(B), r_1(A), r_4(E), r_3(D), C_4$$

↳ NO PUEDE LEER SI YA COMITÉO

Cada commit está después de 9 escrituras (en las transacciones que escriben) y se cumple que

$$C_3 < r_4(B) \quad C_1 < r_2(E) \quad C_1 < r_4(E)$$

⇒ es ACA

PERO NO ES LEGAL





Al estar en H

$$w_1(E) < r_2(E) < r_1(A)$$

un <sup>de E</sup>unlecho luego de  $r_1(A)$

fuerza a que

$$r_1(A) < r_2(E)$$

la cual no corresponde con H ✓





b) i) Para que un planificador multiversion haga rollback tiene que pasar que  ~~$TS(T) < R(A)$~~

$TS(T) < R(A)$  al ~~querer~~ querer efectuar una lectura

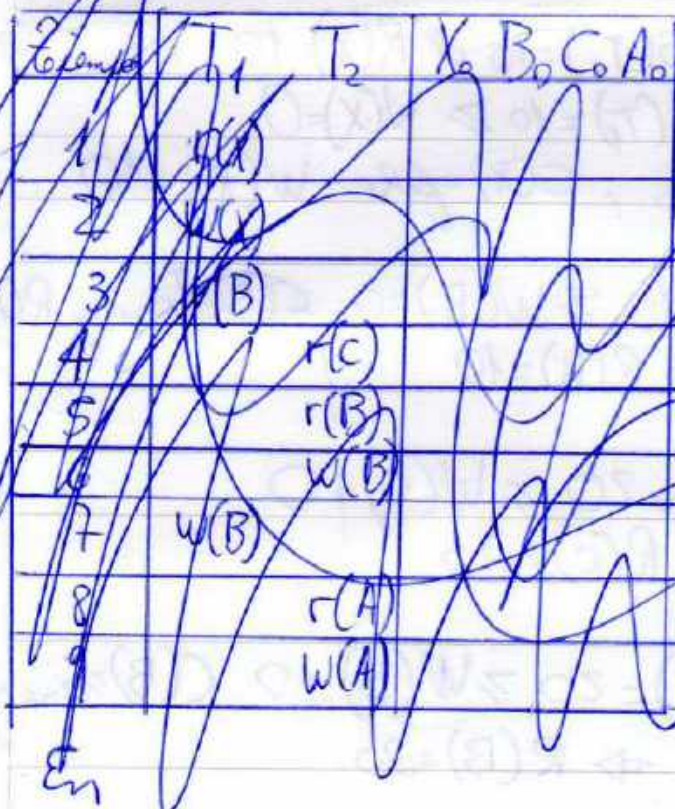
No que el único recurso con el que ambas transacciones interactúan es "B". Entonces voy a hacer que ~~las primeras~~ suceda esto:

$r_1(B)$   $r_2(B)$   $w_2(B)$   $w_1(B)$

va a dejar  
 $R(B) = TS(T_1)$   $R(B) = TS(T_2)$   
 si  $TS(T_1) < TS(T_2)$

podría escribir

$TS(T_1) < R(B) = TS(T_2)$   
 esto produce el rollback





	$T_1$	$T_2$	$X_{10}$	$B_0$	$C_0$	$A_0$	$X_{10}$
1)	$ST=10$						
2)		$ST=20$					
3)	$r(X)$						
4)	$w(X)$						
5)	$r(B)$						
6)		$r(C)$					
7)		$r(B)$					
8)		$w(B)$					
9)	$w(B)$						
10)		$r(A)$					
11)		$w(A)$					

En 3)  $T_1$  lee  $X$ , como  $TS(T_1)=10 \geq w(X)=0$ , puede hacerla y  $RC(X)=10$

En 4)  $T_1$  escribe  $X$ , como  $TS(T_1)=10 \geq R(X)=10$  y  $TS(T_1)=10 \geq w(X)=0$   
entonces se crea  $X_{10}$ ;  $C(X)=false$   $w(X)=10$

En 5)  $T_1$  lee  $\underline{X}$   $B$   $TS(T_1)=10 \geq w(B)=0$   $C(B)=true$  y  $R(B)=0$   
 $\Rightarrow R(B)=10$

En 6)  $T_2$  lee  $C$   $TS(T_2)=20 \geq w(C)=0$   
 $R(C)=20$

En 7)  $T_2$  lee  $B$   $TS(T_2)=20 \geq w(B)=0$   $C(B)=true$  y  $R(B)=10$   
 $\Rightarrow R(B)=20$

En 8)  $T_2$  escribe  $B$   $TS(T_2)=20 \geq R(B)=20$  y  $TS(T_2)=20 \geq w(B)=10$   
 $\Rightarrow C(B)=false$  y  $w(B)=20$



En 9)

$$TS(T_1) = 10 < R(B) = 20$$

$\Rightarrow T_1$  aborta y hace rollback

se deshace lo que hizo

$$W(X) = 0$$

$$R(X) = 0$$

$$C(X) = \text{true}$$

X vuelve a su antiguo valor

Intenta  
escribir

En 10)  $T_2$  lee A  $TS(T_2) = 20 \geq W(A) = 0$   $C(A) = \text{true}$   
 $\Rightarrow R(A) = 20$   $TS(T_2) = 20 \geq 0 = R(A)$

En 11)  $T_2$  escribe A

$$TS(T_2) = 20 \geq R(A) = 20$$

$$TS(T_2) = 20 \geq W(A) = 0$$

$$\Rightarrow W(A) = 20 \quad C(A) = \text{false}$$

Luego ~~se~~

$T_2$  commita con lo cual

$$C(A) = \text{true} \text{ y } C(B) = \text{true}$$



i()	$T_1$	$T_2$
1		$ST=20$
2	$ST=30$	
3	$r(X)$	
4	$w(X)$	
5	$r(B)$	
6		$r(C)$
7		$r(B)$
8		$w(B)$
9	$w(B)$	
10		$r(A)$
11		$w(A)$

En 3  $R(X) \leftarrow 30$  (primera lectura)

En 4  $w$  puede escribir  $TS(T_1)=30 \not\geq 30=R(X)$  y  
 $30 \not\geq 0=W(X)$

$\Rightarrow W(X)=30$   $C(X)=false$

En 5  $R(B) \leftarrow 30$  (primera lectura)

En 6  $R(C) \leftarrow 20$  (primera lectura)

En 7  $TS(T_2)=20 \not\geq W(B)=0$   
 $C(B)=true$  pero  $TS(T_2) < 30=R(B)$   
 $\Rightarrow$  no hace nada, rollback

En 8

$TS(T_2)=20 < R(B)=30$

$\Rightarrow T_2$  aborta, rollback  
 $R(C) \leftarrow 0$

En 9  $TS(T_1)=30 \not\geq R(B)=30$   $TS(T_1)=30 \not\geq W(B)$

$\Rightarrow$  escribe B  $W(B)=30$

Luego,  $T_1$  commita, con lo cual  $C(B)=true$  y  $C(X)=true$   $C(B)=false$  ✓



c) i) 10  $\langle \text{START CKPT (?)} \rangle$



$\langle \text{START CKPT}(T_2, T_3, T_4) \rangle$

ya que  $T_2, T_3$  y  $T_4$  están activas  
( $T_1$  committed)

21  $\langle \text{START CKPT (?)} \rangle$

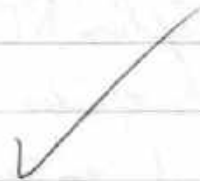


$\langle \text{START CKPT}(T_3, T_5, T_6) \rangle$

$T_3, T_5$  y  $T_6$  están activas ( $T_1, T_2$  y  $T_4$  committed)

$\langle \text{END CKP} \rangle$  PUEDE ocurrir antes que  $\langle \text{COMMIT } T_4 \rangle$   
ya que para finalizar el checkpoint lo que se espera es que se  
haya admitido todo lo ocurrido antes del  $\langle \text{START CKPT} \dots \rangle$   
en la estrategia UNDO/REDO

~~Walter~~



ii) [1] Recorro hacia el  $\langle \text{START CKPT} \dots \rangle$  con  
 $\langle \text{END CKPT} \rangle$  más cercano, anotando las  $T_i$  commitadas  
y las incompletas que estaban activas e iniciando después de ese START

En este caso en 10  $\langle \text{START CKPT}(T_2, T_3, T_4) \rangle$

$T_2$  commitada

$T_4$  commitada

$T_3, T_5$  y  $T_6$  están incompletas

Aplico UNDO sobre  $T_3, T_5$  y  $T_6$

- Por 22  $I \leftarrow 2$

Escribo en disco  $C=20$ ;

- Por 20  $I \leftarrow 50$

$E=40$ ;  $G=20$ ;  $I=50$

- Por 19  $G \leftarrow 3$

- Por 15  $G \leftarrow 20$

Escribo en el log y luego log

- Por 11  $E \leftarrow 40$

a disco  $\langle \text{ABORT } T_3 \rangle$

- Por 7  $C \leftarrow 20$

$\langle \text{ABORT } T_5 \rangle$

$\langle \text{ABORT } T_6 \rangle$

Aplico REDO sobre las commitadas, con ~~los~~ los registros  
ocurridos luego del  $\langle \text{START CKPT}(T_2, T_3, T_4) \rangle$   
(ya que los anteriores están en disco)

- Por 13  $B \leftarrow 12$  Escribo en disco  $B=12$

